EXHIBIT 3

Invalidity of U.S. Patent No. 5,796,967 under 35 U.S.C. § 102 and/or § 103 by U.S. Patent No. 4,688,167 ("Agarwal")[1]

U.S. Patent No. 4,688,167 ("Agarwal"), filed on September 27, 1984 and issued on August 18, 1987, qualifies as prior art to U.S. Patent No. 5,796,967 ("'967 patent") under 35 U.S.C. § 102(a), (b), and/or (e) and anticipates and/or renders obvious one or more claims of the '967 patent.

As described in the following claim chart, one or more claims of the '967 patent are invalid as expressly and inherently anticipated by Agarwal, at least under Plaintiff's apparent constructions.

In addition, to the extent that Agarwal is found not to anticipate, expressly or inherently, one or more of the asserted claims of the '967 patent, these claims are invalid as obvious in view of Agarwal alone or in combination with other prior art references, including but not limited to the prior art identified in the Defendants' Preliminary Invalidity Contentions and the prior art described in the claim charts attached thereto.

Defendants reserve the right to amend or supplement this claim chart at a later date as more fully set forth in the Preliminary Invalidity Contentions.

Claims of the '967 patent	Agarwal
Claim 1	
[1.1] A method for presenting interactive applications on a computer network, the network including a multiplicity of user reception systems at which respective users may request a	Agarwal discloses a method for presenting interactive applications on a computer network, the network including a multiplicity of user reception systems at which respective users may request a multiplicity of available applications. *See, e.g.,:* However, when a task requires little user input the keyboard remains idle. More sophisticated systems allow for multi-tasking. In such systems, an application task which requires little or

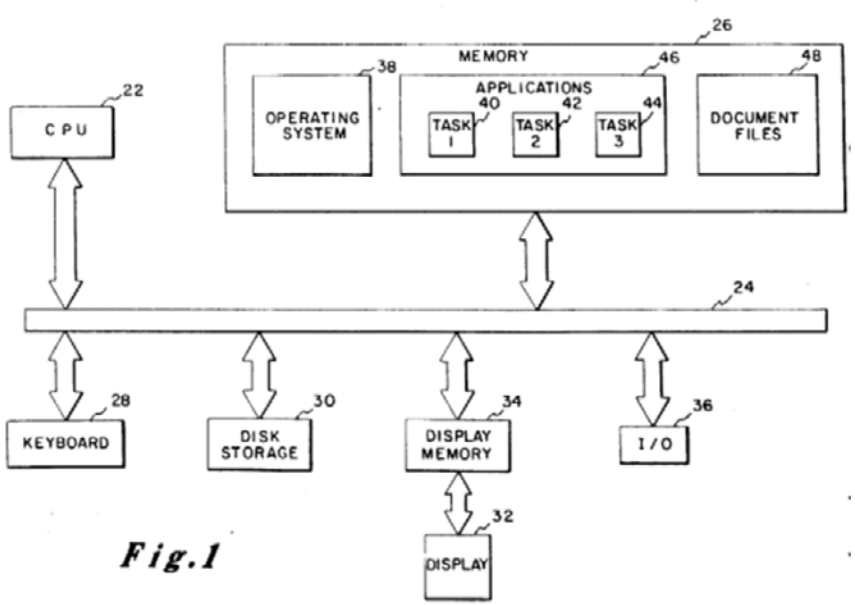
[1] The use of this reference or combinations of references as invalidating prior art under 35 U.S.C. §§ 102 and/or 103 may be based on Plaintiff's allegations of infringement. Defendants do not necessarily agree with the interpretations set forth in Plaintiff's preliminary contentions and thus these invalidity contentions are not an admission that the accused products meet any particular claim element or infringe these claims. Moreover, nothing in these contentions should be interpreted as an acquiescence to or assertion of a particular claim construction by Defendants. In addition, nothing in these contentions should be interpreted as a position about whether any portion of the asserted claims is limiting or not. Further, by submitting these invalidity contentions, Defendants do not waive and hereby expressly reserve their right to raise other invalidity defenses, including but not limited to defenses under 35 U.S.C. §§ 112.

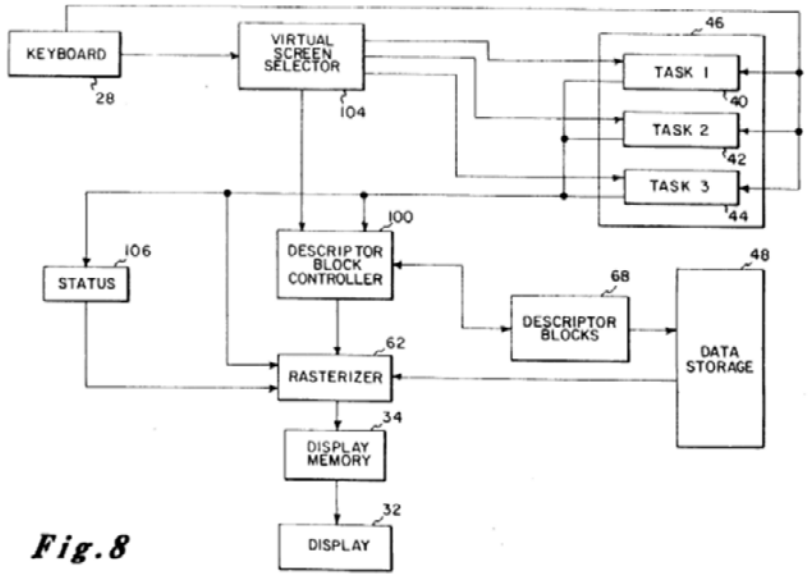
Claims of the ’967 patent	Agarwal
multiplicity of available applications,	no user input is performed by the system in a back ground mode; that is, the task does not interact with the keyboard and leaves the keyboard available to other tasks. A foreground task, on the other hand, which does require user input, interacts with the keyboard. A common display technique for multi-tasking systems is referred to as windowing. In that technique, a document or a portion of a document being processed by the foreground task is predominantly displayed on the system display. Background documents relating to the background tasks are displayed in part so as to be perceived as being positioned below the foreground document but in partial view of the user. A background document can be moved into the foreground by positioning the display cursor over the selected background document. Only the task associated with the foreground document has access to the keyboard. In another form of windowing, displays of documents associated with the various tasks are not overlapped. Rather, the various task windows are positioned in a side-by-side relationship. [1:28-52] The present invention relates to a data processing system having a central processing unit (CPU) which is controlled through an operating system program and application tasks software. Preferably, both the operating system and the application tasks are in the form of software which is loaded into a memory associated with the CPU. [1:55-61] In accordance with one aspect of the invention, the CPU is able to process multiple application tasks together. A screen manager in the operating system is responsive to a plurality of application tasks to designate a plurality of virtual screens, all corresponding to the same single portion of the physical display screen. [1:63-68] The screen manager is also responsive to an input to the data processing system, such as a keyboard input, to select one of the virtual screens for display at the single portion of the physical display screen under control of an application task. Further, the screen manager controls display of identifiers at a second portion of the physical display screen. The

Claims of the ’967 patent	Agarwal
	identifiers correspond to the several virtual screens. Each identifier displayed in the second portion of the physical display may include an indication as to when an error exists in a particular background application task. [2:1-11] The screen manager responds to the application task to change the designated viewport portions of the physical display screen and thus change the size, position and number of viewports. Also, the screen manager responds to the application task to independently change the logical position of a viewport with respect to the document files and to thus independently change the display of data in each view port. The display may also be updated, through the screen manager, to include changes in the stored data made by the application task. [2:44-54] The viewport technique provides a flexible mechanism by which an application task can display data, most likely taken from different pages in the document files, in a side-by-side relationship. The ability to establish viewports is available to each application task. An application task can itself provide even greater flexibility by allowing for a subdivision of the sections of data, such as pages, which may be displayed in the viewports. Those subdivided areas can be independently controlled by the application task software but, unless modified by an application task, are seen as fixed side-by-side areas by the screen manager. Even further flexibility in the system is obtained by allowing each area to include multiple levels with one type of level including text and another including graphic information and the like. Those levels can be superimposed over each other when displayed in each area. [3:14-31] At least one input/output unit 36 is also connected to the bus 24. The input /output unit 36 includes a communications port for communicating with a printer, other work stations or a main processing unit. Although the present invention is described with respect to a standalone word processing and office automation system, the invention is equally applicable to other systems such as distributed systems. [4:13-19] A display from the foreground task is provided on a major portion of the physical display screen indicated as the task screen 50. Under control of the operating system to be described below, the display on the task screen may be divided into a number of display viewports each

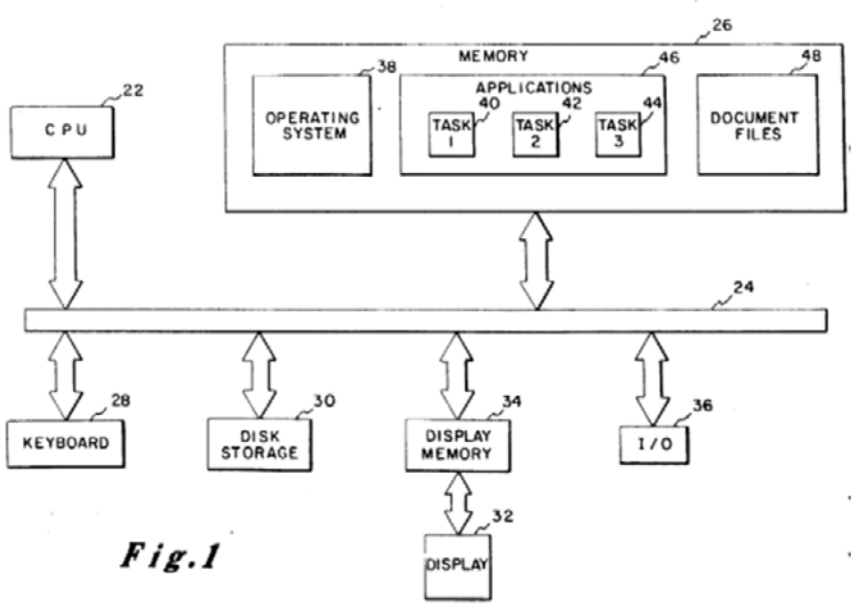
Claims of the ’967 patent	Agarwal
	of which independently displays a different set of information. [4:62-68] It can be seen that the present system offers windowing at two levels. At a task level, in virtual screen windowing a task window covers virtually the entire physical screen. Within each virtual screen established by a particular application task, that task can subdivide the virtual screen into viewport windows. Because each viewport is associated with an active task, the view ports are positioned side-by-side. [5:59-66] The file management system 54 manipulates data to and from the keyboard, disc storage and input /output unit. The file management system interfaces with the peripherals through drivers 56 which include the software required for interfacing with the specific peripherals used. [6:3-8] It can be seen that any number of virtual screens can be established by the screen manager in response to requests from application tasks and each virtual screen can be subdivided into any number of viewports by additional requests from the respective application task. Each virtual screen and each viewport is defined by a descriptor block which sets the size of the virtual screen or viewport, points to a page or document in the document files which is to be displayed in the virtual screen or viewport and sets the logical position of the screen or viewport relative to the page or document. [7:55-65] This controller handles the several functions described above. In particular, it implements the CREATE VIRTUAL SCREEN, CRE ATE VIEWPORT, ASSIGN, UPDATE, MERGE, AND DELETE functions with respect to particular descriptor data blocks 68. [10:3-8] The present technique also allows for the virtual screen of primary interest to make up a much larger portion of the physical screen. The use of the operating screen 52in the display gives the operating system sufficient opportunity to keep the user informed as to the status of virtual screens which are not displayed. Further, the ability of the operating system to establish viewports in each virtual screen greatly adds to the flexibility of the system, particularly with respect to displaying different types of data such as text and graphics. The information

Claims of the ’967 patent	Agarwal
	displayed in different viewports can also be selected from different pages and even different documents in the document files 48. The example of displaying text adjacent to graphics using the view port technique has previously been noted. Establishing viewport descriptor blocks for other items such as the menu and error messages of FIG. 2 makes the screen manager operations extremely flexible. It also minimizes the amount of updating of the screen. For example, in order to update the prompts viewport, which may re quire frequent updating, it is not necessary to as frequently update the entire screen. Similarly, when word processing, it may only be necessary to update the text viewport and not the other viewports at particular stages of an application task. The ability of the applications task to further subdivide pages into areas adds yet another dimension to the control of information to be displayed. It allows the application task to establish areas to be displayed in a relatively fixed relationship as far as the screen manager is concerned; whereas, the viewport technique requires the screen manager to handle each viewport more independently. Establishing areas simplifies certain tasks of the application software such as formatting, wrap around within columns and the like. [10:42-11:8]

Claims of the '967 patent	Agarwal
	 *Fig. 1* [Figure 1]

Claims of the '967 patent	Agarwal
	 *Fig. 8* [Figure 8] Additional prior art that discloses this limitation and is combinable with the present prior art is found in the relevant section of charts for other prior art for the '967 patent. It would have been obvious to one of ordinary skill in the art at the time of the invention to include such functionality in the system disclosed in Agarwal. *See, e.g.*, Preliminary Invalidity Contentions Section II.D.
[1.2] the respective reception systems including a monitor at which the applications requested can be presented as one or more screens of display, the method	Agarwal discloses that the respective reception systems include a monitor at which the applications requested can be presented as one or more screens of display. *See, e.g.,:* This system is also provided with a video display. [1:61-62]

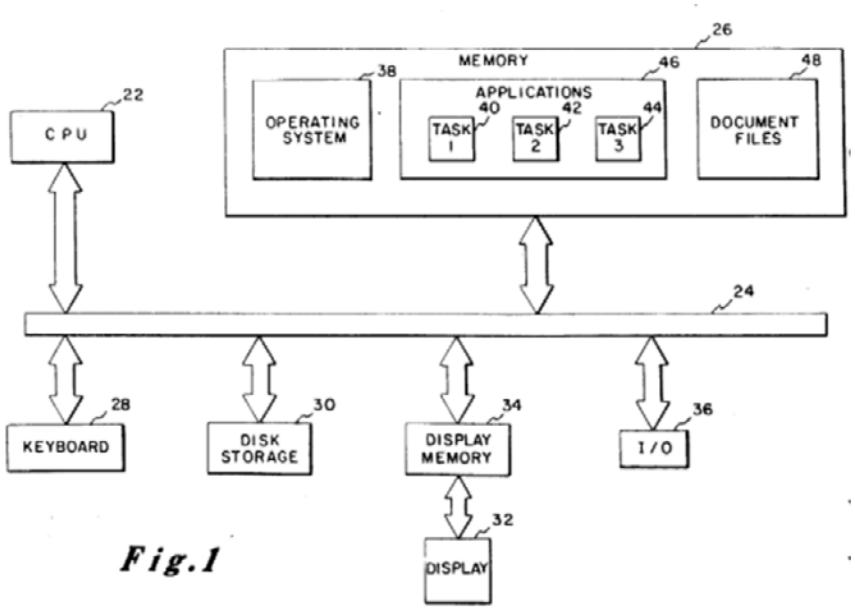
Claims of the ’967 patent	Agarwal
comprising the steps of:	A memory associated with the display may be a bitmap memory which includes individual data corresponding to each pixel of the display. A screen manager system within the operating system may include a software based rasterizer which generates the individual pixel data. [2:28-32] The peripherals include a keyboard 28, a magnetic disc storage unit 30, a display 32 which is preferably a cathode ray tube display and an associated display memory 34. [4:10-13] That data is passed through the rasterizer 62 of the screen manager to generate the signal to be applied at each pixel of the display screen. The code for each pixel is stored in an 800 by 300 bit display memory 34. The screen manager also selects the information to be dis played on the operating system screen 52 designated in a descriptor block 74 and, through the rasterizer 62, stores corresponding pixel information in the memory 34. [8:1-9] From these inputs, the rasterizer generates a complete video display. Updating the display may be made in response to signals from application tasks when the underlying data is changed or in response to changes in descriptor blocks. [10:22-26]

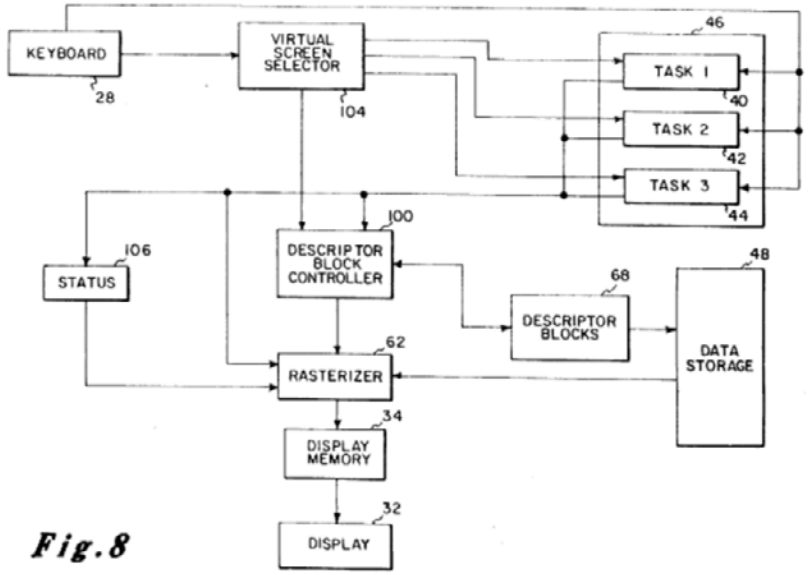
Claims of the '967 patent	Agarwal
	 *Fig. 1* [Figure 1]
[1.3] a. generating a screen display at a respective reception system for a requested application, the screen display being generated by the respective reception system from data objects having a prescribed data structure,	Agarwal discloses generating a screen display at a respective reception system for a requested application, the screen display being generated by the respective reception system from data objects having a prescribed data structure. *See, e.g.,:* However, when a task requires little user input the keyboard remains idle. More sophisticated systems allow for multi-tasking. In such systems, an application task which requires little or no user input is performed by the system in a back ground mode; that is, the task does not interact with the keyboard and leaves the keyboard available to other tasks. A foreground task, on the other hand, which does require user input, interacts with the keyboard.

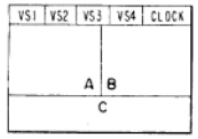
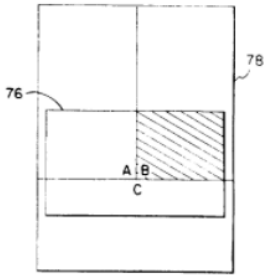
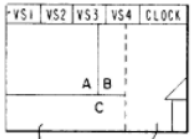
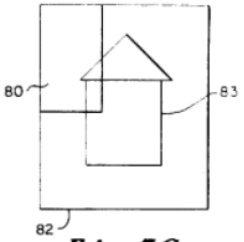
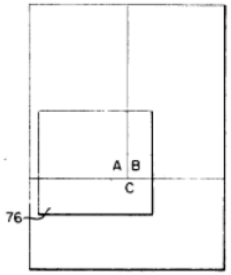
Claims of the '967 patent	Agarwal
	A common display technique for multi-tasking systems is referred to as windowing. In that technique, a document or a portion of a document being processed by the foreground task is predominantly displayed on the system display. Background documents relating to the background tasks are displayed in part so as to be perceived as being positioned below the foreground document but in partial view of the user. A background document can be moved into the foreground by positioning the display cursor over the selected background document. Only the task associated with the foreground document has access to the keyboard. In another form of windowing, displays of documents associated with the various tasks are not overlapped. Rather, the various task windows are positioned in a side-by-side relationship. [1:28-52] The present invention relates to a data processing system having a central processing unit (CPU) which is controlled through an operating system program and application tasks software. Preferably, both the operating system and the application tasks are in the form of software which is loaded into a memory associated with the CPU. [1:55-61] In accordance with one aspect of the invention, the CPU is able to process multiple application tasks together. A screen manager in the operating system is responsive to a plurality of application tasks to designate a plurality of virtual screens, all corresponding to the same single portion of the physical display screen. [1:63-68] The screen manager is also responsive to an input to the data processing system, such as a keyboard input, to select one of the virtual screens for display at the single portion of the physical display screen under control of an application task. Further, the screen manager controls display of identifiers at a second portion of the physical display screen. The identifiers correspond to the several virtual screens. Each identifier displayed in the second portion of the physical display may include an indication as to when an error exists in a particular background application task. [2:1-11]

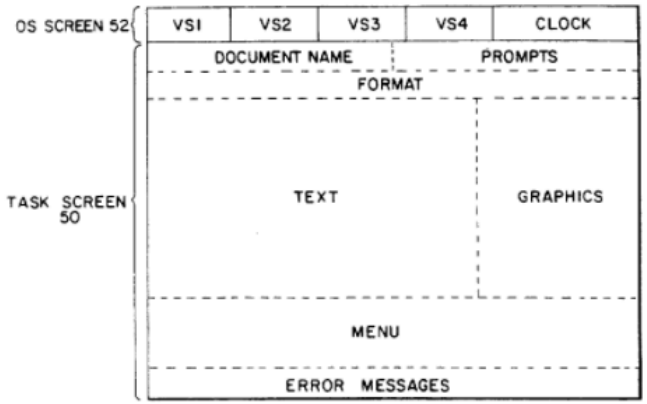
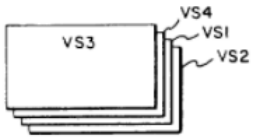
Claims of the '967 patent	Agarwal
	The screen manager responds to the application task to change the designated viewport portions of the physical display screen and thus change the size, position and number of viewports. Also, the screen manager responds to the application task to independently change the logical position of a viewport with respect to the document files and to thus independently change the display of data in each view port. The display may also be updated, through the screen manager, to include changes in the stored data made by the application task. [2:44-54] The viewport technique provides a flexible mechanism by which an application task can display data, most likely taken from different pages in the document files, in a side-by-side relationship. The ability to establish viewports is available to each application task. An application task can itself provide even greater flexibility by allowing for a subdivision of the sections of data, such as pages, which may be displayed in the viewports. Those subdivided areas can be independently controlled by the application task software but, unless modified by an application task, are seen as fixed side-by-side areas by the screen manager. Even further flexibility in the system is obtained by allowing each area to include multiple levels with one type of level including text and another including graphic information and the like. Those levels can be superimposed over each other when displayed in each area. [3:14-31] At least one input/output unit 36 is also connected to the bus 24. The input /output unit 36 includes a communications port for communicating with a printer, other work stations or a main processing unit. Although the present invention is described with respect to a standalone word processing and office automation system, the invention is equally applicable to other systems such as distributed systems. [4:13-19] A display from the foreground task is provided on a major portion of the physical display screen indicated as the task screen 50. Under control of the operating system to be described below, the display on the task screen may be divided into a number of display viewports each of which independently displays a different set of information. [4:62-68] It can be seen that the present system offers windowing at two levels. At a task level, in virtual screen windowing a task window covers virtually the entire physical screen. Within each

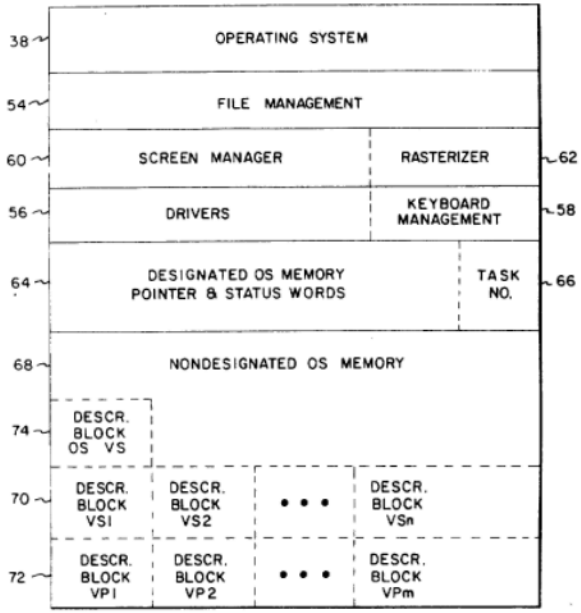
Claims of the ’967 patent	Agarwal
	virtual screen established by a particular application task, that task can subdivide the virtual screen into viewport windows. Because each viewport is associated with an active task, the view ports are positioned side-by-side. [5:59-66] The file management system 54 manipulates data to and from the keyboard, disc storage and input /output unit. The file management system interfaces with the peripherals through drivers 56 which include the software required for interfacing with the specific peripherals used. [6:3-8] It can be seen that any number of virtual screens can be established by the screen manager in response to requests from application tasks and each virtual screen can be subdivided into any number of viewports by additional requests from the respective application task. Each virtual screen and each viewport is defined by a descriptor block which sets the size of the virtual screen or viewport, points to a page or document in the document files which is to be displayed in the virtual screen or viewport and sets the logical position of the screen or viewport relative to the page or document. [7:55-65] This controller handles the several functions described above. In particular, it implements the CREATE VIRTUAL SCREEN, CRE ATE VIEWPORT, ASSIGN, UPDATE, MERGE, AND DELETE functions with respect to particular descriptor data blocks 68. [10:3-8] The present technique also allows for the virtual screen of primary interest to make up a much larger portion of the physical screen. The use of the operating screen 52in the display gives the operating system sufficient opportunity to keep the user informed as to the status of virtual screens which are not displayed. Further, the ability of the operating system to establish viewports in each virtual screen greatly adds to the flexibility of the system, particularly with respect to displaying different types of data such as text and graphics. The information displayed in different viewports can also be selected from different pages and even different documents in the document files 48. The example of displaying text adjacent to graphics using the view port technique has previously been noted. Establishing viewport descriptor blocks for other items such as the menu and error messages of FIG. 2 makes the screen manager

Claims of the ’967 patent	Agarwal
	operations extremely flexible. It also minimizes the amount of updating of the screen. For example, in order to update the prompts viewport, which may re quire frequent updating, it is not necessary to as frequently update the entire screen. Similarly, when word processing, it may only be necessary to update the text viewport and not the other viewports at particular stages of an application task. The ability of the applications task to further subdivide pages into areas adds yet another dimension to the control of information to be displayed. It allows the application task to establish areas to be displayed in a relatively fixed relationship as far as the screen manager is concerned; whereas, the viewport technique requires the screen manager to handle each viewport more independently. Establishing areas simplifies certain tasks of the application software such as formatting, wrap around within columns and the like. [10:42-11:8]  *Fig. 1*

Claims of the ’967 patent	Agarwal
	[Figure 1]  *Fig. 8* [Figure 8]

Claims of the ’967 patent	Agarwal
	 *Fig.6A*  *Fig.6B*  *Fig.7A*  *Fig.7C*  *Fig.7B* [Figures 6-7]

Claims of the ’967 patent	Agarwal
	 *Fig. 2*  *Fig. 3* [Figures 2-3]

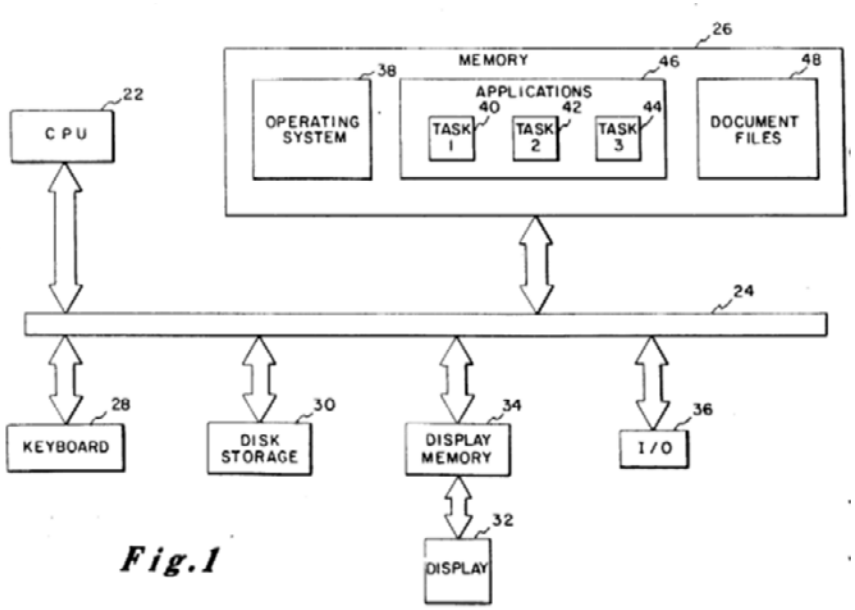
Claims of the '967 patent	Agarwal
	 *Fig.4* [Figure 4] Additional prior art that discloses this limitation and is combinable with the present prior art is found in the relevant section of charts for other prior art for the '967 patent. It would have been obvious to one of ordinary skill in the art at the time of the invention to include such functionality in the system disclosed in Agarwal. *See, e.g.*, Preliminary Invalidity Contentions Section II.D.
[1.4] at least some of which objects may be stored at the	Agarwal discloses that at least some of which objects may be stored at the respective reception system. *See, e.g.,:*

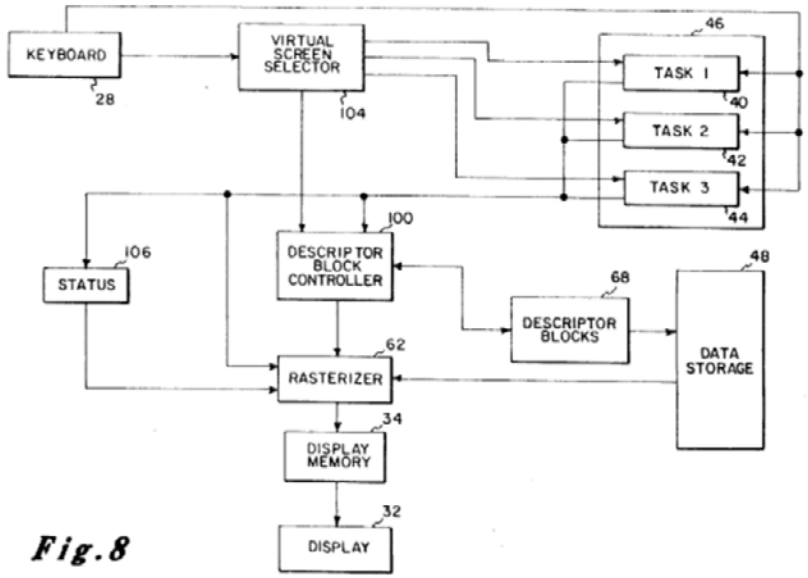
Claims of the '967 patent	Agarwal
respective reception system,	Distinct sections of document data stored in memory under control of the applications task are designated for each viewport. [2:38-40.] During start-up of this system, the operating software 38 is loaded into the memory 26 from the disc storage 30. That software, the operating system, controls the general operation of the CPU and the associated peripherals and serves as an interface between the CPU and peripherals and the applications software. Once the system is running under control of the operating system, the system user may select, through the keyboard 28, any of a number of application software packages from disc storage 30 and load them into the memory 26. In the illustration of FIG. 1, three independent application packages 40, 42 and 44 have been loaded into the memory at 46. [4:21-33] The file management system 54 manipulates data to and from the keyboard, disc storage and input /output unit. The file management system interfaces with the peripherals through drivers 56 which include the software required for interfacing with the specific peripherals used. [6:3-8]
[1.5] the screen display including a plurality of partitions, the partitions being constructed from objects, the objects being retrieved from the objects stored at the respective reception system, or if unavailable from the objects stored at the respective reception system, then from the network, such that at least some of the objects may be used in more than one application;	Agarwal discloses that the screen display includes a plurality of partitions, the partitions being constructed from objects, the objects being retrieved from the objects stored at the respective reception system, or if unavailable from the objects stored at the respective reception system, then from the network, such that at least some of the objects may be used in more than one application. *See, e.g.,:* However, when a task requires little user input the keyboard remains idle. More sophisticated systems allow for multi-tasking. In such systems, an application task which requires little or no user input is performed by the system in a back ground mode; that is, the task does not interact with the keyboard and leaves the keyboard available to other tasks. A foreground task, on the other hand, which does require user input, interacts with the keyboard. A common display technique for multi-tasking systems is referred to as windowing. In that

Claims of the '967 patent	Agarwal
	technique, a document or a portion of a document being processed by the foreground task is predominantly displayed on the system display. Background documents relating to the background tasks are displayed in part so as to be perceived as being positioned below the foreground document but in partial view of the user. A background document can be moved into the foreground by positioning the display cursor over the selected background document. Only the task associated with the foreground document has access to the keyboard. In another form of windowing, displays of documents associated with the various tasks are not overlapped. Rather, the various task windows are positioned in a side-by-side relationship. [1:28-52] The present invention relates to a data processing system having a central processing unit (CPU) which is controlled through an operating system program and application tasks software. Preferably, both the operating system and the application tasks are in the form of software which is loaded into a memory associated with the CPU. [1:55-61] In accordance with one aspect of the invention, the CPU is able to process multiple application tasks together. A screen manager in the operating system is responsive to a plurality of application tasks to designate a plurality of virtual screens, all corresponding to the same single portion of the physical display screen. [1:63-68] The screen manager is also responsive to an input to the data processing system, such as a keyboard input, to select one of the virtual screens for display at the single portion of the physical display screen under control of an application task. Further, the screen manager controls display of identifiers at a second portion of the physical display screen. The identifiers correspond to the several virtual screens. Each identifier displayed in the second portion of the physical display may include an indication as to when an error exists in a particular background application task. [2:1-11] The screen manager responds to the application task to change the designated viewport

Claims of the ’967 patent	Agarwal
	portions of the physical display screen and thus change the size, position and number of viewports. Also, the screen manager responds to the application task to independently change the logical position of a viewport with respect to the document files and to thus independently change the display of data in each view port. The display may also be updated, through the screen manager, to include changes in the stored data made by the application task. [2:44-54] The viewport technique provides a flexible mechanism by which an application task can display data, most likely taken from different pages in the document files, in a side-by-side relationship. The ability to establish viewports is available to each application task. An application task can itself provide even greater flexibility by allowing for a subdivision of the sections of data, such as pages, which may be displayed in the viewports. Those subdivided areas can be independently controlled by the application task software but, unless modified by an application task, are seen as fixed side-by-side areas by the screen manager. Even further flexibility in the system is obtained by allowing each area to include multiple levels with one type of level including text and another including graphic information and the like. Those levels can be superimposed over each other when displayed in each area. [3:14-31] At least one input/output unit 36 is also connected to the bus 24. The input /output unit 36 includes a communications port for communicating with a printer, other work stations or a main processing unit. Although the present invention is described with respect to a standalone word processing and office automation system, the invention is equally applicable to other systems such as distributed systems. [4:13-19] A display from the foreground task is provided on a major portion of the physical display screen indicated as the task screen 50. Under control of the operating system to be described below, the display on the task screen may be divided into a number of display viewports each of which independently displays a different set of information. [4:62-68] It can be seen that the present system offers windowing at two levels. At a task level, in virtual screen windowing a task window covers virtually the entire physical screen. Within each virtual screen established by a particular application task, that task can subdivide the virtual

Claims of the '967 patent	Agarwal
	screen into viewport windows. Because each viewport is associated with an active task, the view ports are positioned side-by-side. [5:59-66] The file management system 54 manipulates data to and from the keyboard, disc storage and input /output unit. The file management system interfaces with the peripherals through drivers 56 which include the software required for interfacing with the specific peripherals used. [6:3-8] It can be seen that any number of virtual screens can be established by the screen manager in response to requests from application tasks and each virtual screen can be subdivided into any number of viewports by additional requests from the respective application task. Each virtual screen and each viewport is defined by a descriptor block which sets the size of the virtual screen or viewport, points to a page or document in the document files which is to be displayed in the virtual screen or viewport and sets the logical position of the screen or viewport relative to the page or document. [7:55-65] This controller handles the several functions described above. In particular, it implements the CREATE VIRTUAL SCREEN, CRE ATE VIEWPORT, ASSIGN, UPDATE, MERGE, AND DELETE functions with respect to particular descriptor data blocks 68. [10:3-8] The present technique also allows for the virtual screen of primary interest to make up a much larger portion of the physical screen. The use of the operating screen 52 in the display gives the operating system sufficient opportunity to keep the user informed as to the status of virtual screens which are not displayed. Further, the ability of the operating system to establish viewports in each virtual screen greatly adds to the flexibility of the system, particularly with respect to displaying different types of data such as text and graphics. The information displayed in different viewports can also be selected from different pages and even different documents in the document files 48. The example of displaying text adjacent to graphics using the view port technique has previously been noted. Establishing viewport descriptor blocks for other items such as the menu and error messages of FIG. 2 makes the screen manager operations extremely flexible. It also minimizes the amount of updating of the screen. For

Claims of the ’967 patent	Agarwal
	example, in order to update the prompts viewport, which may re quire frequent updating, it is not necessary to as frequently update the entire screen. Similarly, when word processing, it may only be necessary to update the text viewport and not the other viewports at particular stages of an application task. The ability of the applications task to further subdivide pages into areas adds yet another dimension to the control of information to be displayed. It allows the application task to establish areas to be displayed in a relatively fixed relationship as far as the screen manager is concerned; whereas, the viewport technique requires the screen manager to handle each viewport more independently. Establishing areas simplifies certain tasks of the application software such as formatting, wrap around within columns and the like. [10:42-11:8]  *Fig. 1* [Figure 1]

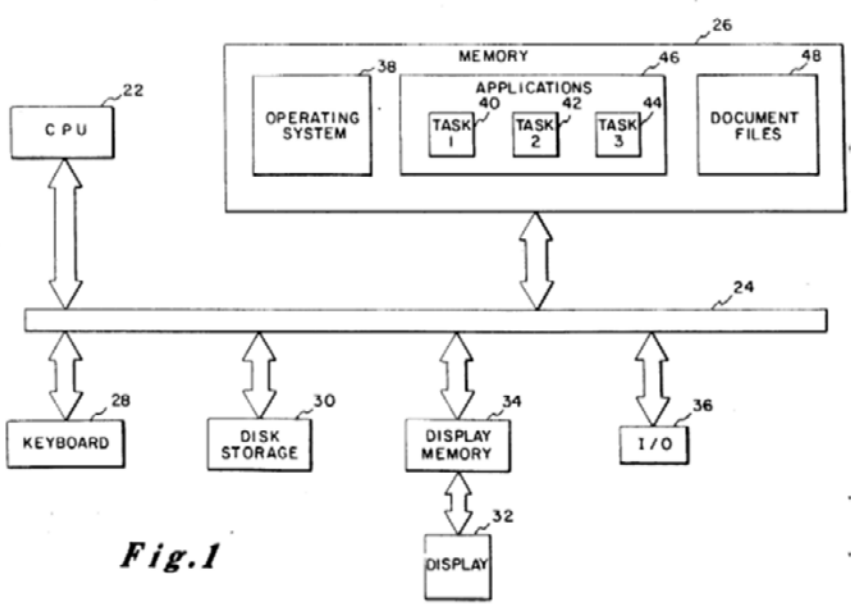
Claims of the '967 patent	Agarwal
	 *Fig. 8* [Figure 8] Additional prior art that discloses this limitation and is combinable with the present prior art is found in the relevant section of charts for other prior art for the '967 patent. It would have been obvious to one of ordinary skill in the art at the time of the invention to include such functionality in the system disclosed in Agarwal. *See, e.g.*, Preliminary Invalidity Contentions Section II.D.
[1.6] b. generating at least a first partition for presenting applications; and	Agarwal discloses generating at least a first partition for presenting applications. *See, e.g.,:* However, when a task requires little user input the keyboard remains idle. More sophisticated systems allow for multi-tasking. In such systems, an application task which requires little or

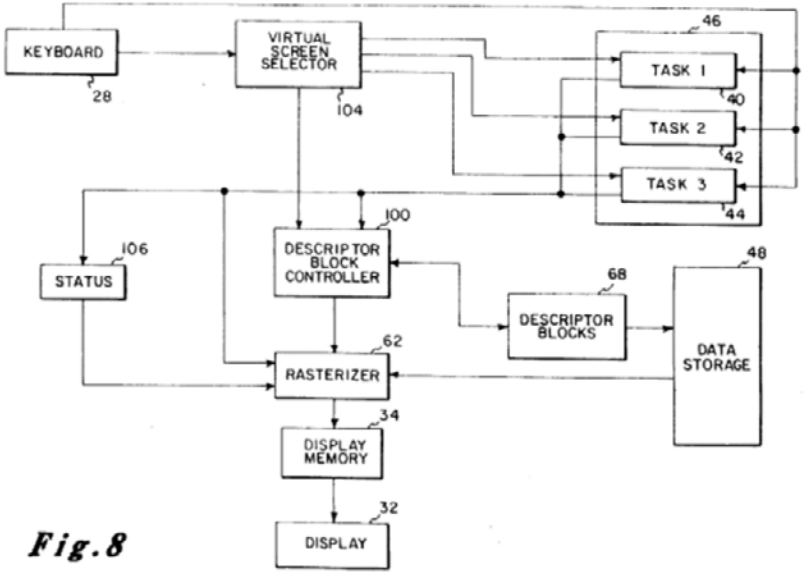
Claims of the ’967 patent	Agarwal
	no user input is performed by the system in a back ground mode; that is, the task does not interact with the keyboard and leaves the keyboard available to other tasks. A foreground task, on the other hand, which does require user input, interacts with the keyboard. A common display technique for multi-tasking systems is referred to as windowing. In that technique, a document or a portion of a document being processed by the foreground task is predominantly displayed on the system display. Background documents relating to the background tasks are displayed in part so as to be perceived as being positioned below the foreground document but in partial view of the user. A background document can be moved into the foreground by positioning the display cursor over the selected background document. Only the task associated with the foreground document has access to the keyboard. In another form of windowing, displays of documents associated with the various tasks are not overlapped. Rather, the various task windows are positioned in a side-by-side relationship. [1:28-52] The present invention relates to a data processing system having a central processing unit (CPU) which is controlled through an operating system program and application tasks software. Preferably, both the operating system and the application tasks are in the form of software which is loaded into a memory associated with the CPU. [1:55-61] In accordance with one aspect of the invention, the CPU is able to process multiple application tasks together. A screen manager in the operating system is responsive to a plurality of application tasks to designate a plurality of virtual screens, all corresponding to the same single portion of the physical display screen. [1:63-68] The screen manager is also responsive to an input to the data processing system, such as a keyboard input, to select one of the virtual screens for display at the single portion of the physical display screen under control of an application task. Further, the screen manager controls display of identifiers at a second portion of the physical display screen. The

Claims of the ’967 patent	Agarwal
	identifiers correspond to the several virtual screens. Each identifier displayed in the second portion of the physical display may include an indication as to when an error exists in a particular background application task. [2:1-11] The screen manager responds to the application task to change the designated viewport portions of the physical display screen and thus change the size, position and number of viewports. Also, the screen manager responds to the application task to independently change the logical position of a viewport with respect to the document files and to thus independently change the display of data in each view port. The display may also be updated, through the screen manager, to include changes in the stored data made by the application task. [2:44-54] The viewport technique provides a flexible mechanism by which an application task can display data, most likely taken from different pages in the document files, in a side-by-side relationship. The ability to establish viewports is available to each application task. An application task can itself provide even greater flexibility by allowing for a subdivision of the sections of data, such as pages, which may be displayed in the viewports. Those subdivided areas can be independently controlled by the application task software but, unless modified by an application task, are seen as fixed side-by-side areas by the screen manager. Even further flexibility in the system is obtained by allowing each area to include multiple levels with one type of level including text and another including graphic information and the like. Those levels can be superimposed over each other when displayed in each area. [3:14-31] At least one input/output unit 36 is also connected to the bus 24. The input /output unit 36 includes a communications port for communicating with a printer, other work stations or a main processing unit. Although the present invention is described with respect to a standalone word processing and office automation system, the invention is equally applicable to other systems such as distributed systems. [4:13-19] A display from the foreground task is provided on a major portion of the physical display screen indicated as the task screen 50. Under control of the operating system to be described below, the display on the task screen may be divided into a number of display viewports each

Claims of the '967 patent	Agarwal
	of which independently displays a different set of information. [4:62-68] It can be seen that the present system offers windowing at two levels. At a task level, in virtual screen windowing a task window covers virtually the entire physical screen. Within each virtual screen established by a particular application task, that task can subdivide the virtual screen into viewport windows. Because each viewport is associated with an active task, the view ports are positioned side-by-side. [5:59-66] The file management system 54 manipulates data to and from the keyboard, disc storage and input /output unit. The file management system interfaces with the peripherals through drivers 56 which include the software required for interfacing with the specific peripherals used. [6:3-8] It can be seen that any number of virtual screens can be established by the screen manager in response to requests from application tasks and each virtual screen can be subdivided into any number of viewports by additional requests from the respective application task. Each virtual screen and each viewport is defined by a descriptor block which sets the size of the virtual screen or viewport, points to a page or document in the document files which is to be displayed in the virtual screen or viewport and sets the logical position of the screen or viewport relative to the page or document. [7:55-65] This controller handles the several functions described above. In particular, it implements the CREATE VIRTUAL SCREEN, CRE ATE VIEWPORT, ASSIGN, UPDATE, MERGE, AND DELETE functions with respect to particular descriptor data blocks 68. [10:3-8] The present technique also allows for the virtual screen of primary interest to make up a much larger portion of the physical screen. The use of the operating screen 52in the display gives the operating system sufficient opportunity to keep the user informed as to the status of virtual screens which are not displayed. Further, the ability of the operating system to establish viewports in each virtual screen greatly adds to the flexibility of the system, particularly with respect to displaying different types of data such as text and graphics. The information

Claims of the ’967 patent	Agarwal
	displayed in different viewports can also be selected from different pages and even different documents in the document files 48. The example of displaying text adjacent to graphics using the view port technique has previously been noted. Establishing viewport descriptor blocks for other items such as the menu and error messages of FIG. 2 makes the screen manager operations extremely flexible. It also minimizes the amount of updating of the screen. For example, in order to update the prompts viewport, which may re quire frequent updating, it is not necessary to as frequently update the entire screen. Similarly, when word processing, it may only be necessary to update the text viewport and not the other viewports at particular stages of an application task. The ability of the applications task to further subdivide pages into areas adds yet another dimension to the control of information to be displayed. It allows the application task to establish areas to be displayed in a relatively fixed relationship as far as the screen manager is concerned; whereas, the viewport technique requires the screen manager to handle each viewport more independently. Establishing areas simplifies certain tasks of the application software such as formatting, wrap around within columns and the like. [10:42-11:8]

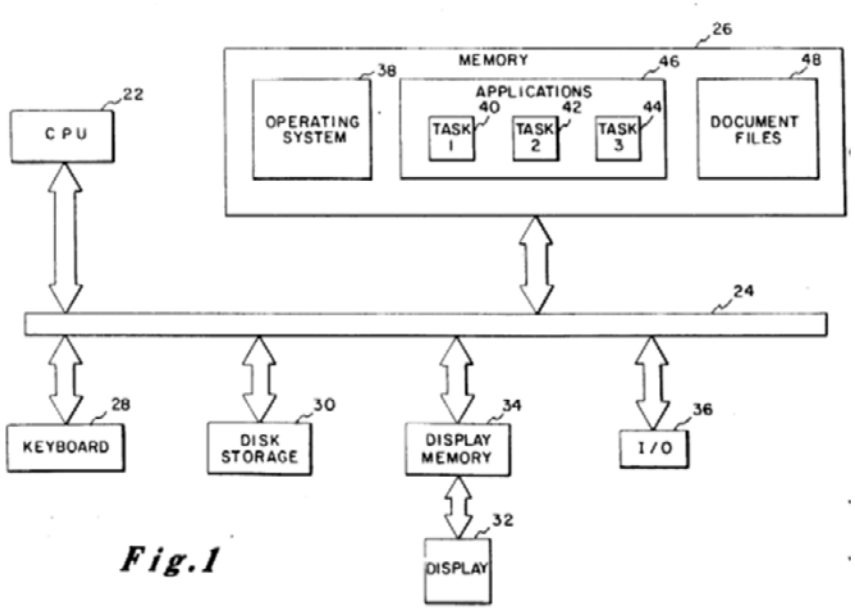
Claims of the '967 patent	Agarwal
	 *Fig. 1* [Figure 1]

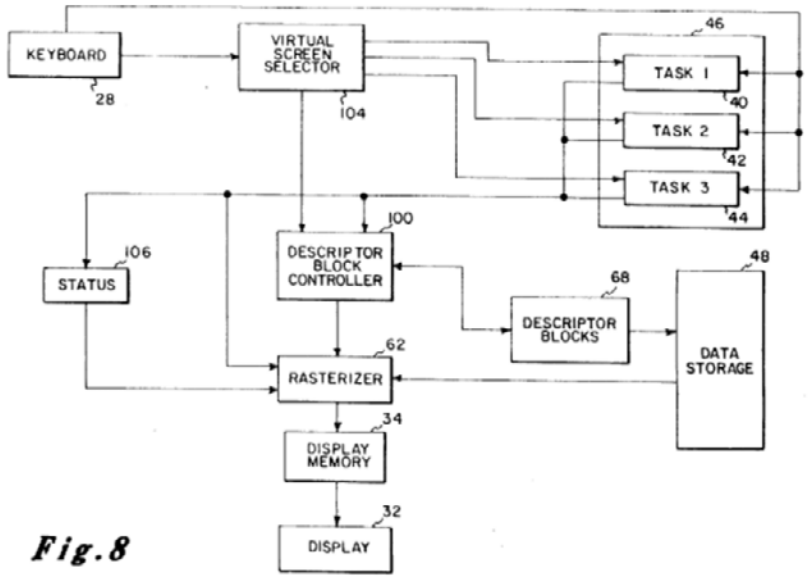
Claims of the '967 patent	Agarwal
	 *Fig. 8* [Figure 8]
[1.7] c. generating concurrently with the first partition at least a second partition for presenting a plurality of command functions, the command functions including at least a first group which are selectable to permit movement between applications.	Agarwal discloses generating concurrently with the first partition at least a second partition for presenting a plurality of command functions, the command functions including at least a first group which are selectable to permit movement between applications. *See, e.g.,:* However, when a task requires little user input the keyboard remains idle. More sophisticated systems allow for multi-tasking. In such systems, an application task which requires little or no user input is performed by the system in a back ground mode; that is, the task does not interact with the keyboard and leaves the keyboard available to other tasks. A foreground task, on the other hand, which does require user input, interacts with the keyboard. A common display technique for multi-tasking systems is referred to as windowing. In that

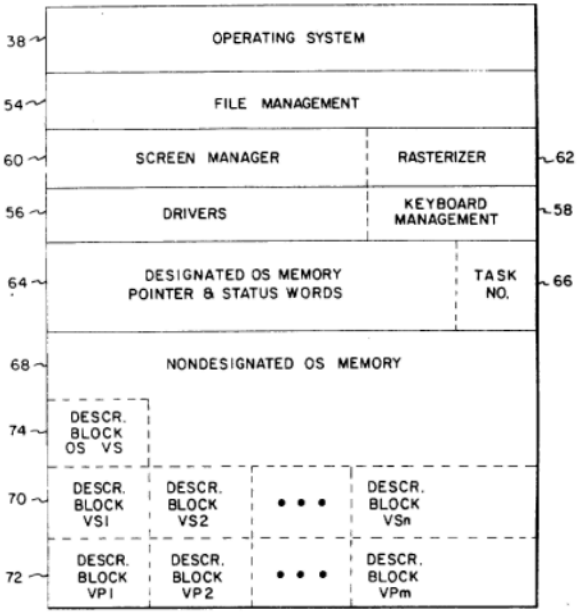
Claims of the ’967 patent	Agarwal
	technique, a document or a portion of a document being processed by the foreground task is predominantly displayed on the system display. Background documents relating to the background tasks are displayed in part so as to be perceived as being positioned below the foreground document but in partial view of the user. A background document can be moved into the foreground by positioning the display cursor over the selected background document. Only the task associated with the foreground document has access to the keyboard. In another form of windowing, displays of documents associated with the various tasks are not overlapped. Rather, the various task windows are positioned in a side-by-side relationship. [1:28-52] The present invention relates to a data processing system having a central processing unit (CPU) which is controlled through an operating system program and application tasks software. Preferably, both the operating system and the application tasks are in the form of software which is loaded into a memory associated with the CPU. [1:55-61] In accordance with one aspect of the invention, the CPU is able to process multiple application tasks together. A screen manager in the operating system is responsive to a plurality of application tasks to designate a plurality of virtual screens, all corresponding to the same single portion of the physical display screen. [1:63-68] The screen manager is also responsive to an input to the data processing system, such as a keyboard input, to select one of the virtual screens for display at the single portion of the physical display screen under control of an application task. Further, the screen manager controls display of identifiers at a second portion of the physical display screen. The identifiers correspond to the several virtual screens. Each identifier displayed in the second portion of the physical display may include an indication as to when an error exists in a particular background application task. [2:1-11] The screen manager responds to the application task to change the designated viewport

Claims of the ’967 patent	Agarwal
	portions of the physical display screen and thus change the size, position and number of viewports. Also, the screen manager responds to the application task to independently change the logical position of a viewport with respect to the document files and to thus independently change the display of data in each view port. The display may also be updated, through the screen manager, to include changes in the stored data made by the application task. [2:44-54] The viewport technique provides a flexible mechanism by which an application task can display data, most likely taken from different pages in the document files, in a side-by-side relationship. The ability to establish viewports is available to each application task. An application task can itself provide even greater flexibility by allowing for a subdivision of the sections of data, such as pages, which may be displayed in the viewports. Those subdivided areas can be independently controlled by the application task software but, unless modified by an application task, are seen as fixed side-by-side areas by the screen manager. Even further flexibility in the system is obtained by allowing each area to include multiple levels with one type of level including text and another including graphic information and the like. Those levels can be superimposed over each other when displayed in each area. [3:14-31] At least one input/output unit 36 is also connected to the bus 24. The input /output unit 36 includes a communications port for communicating with a printer, other work stations or a main processing unit. Although the present invention is described with respect to a standalone word processing and office automation system, the invention is equally applicable to other systems such as distributed systems. [4:13-19] A display from the foreground task is provided on a major portion of the physical display screen indicated as the task screen 50. Under control of the operating system to be described below, the display on the task screen may be divided into a number of display viewports each of which independently displays a different set of information. [4:62-68] It can be seen that the present system offers windowing at two levels. At a task level, in virtual screen windowing a task window covers virtually the entire physical screen. Within each virtual screen established by a particular application task, that task can subdivide the virtual

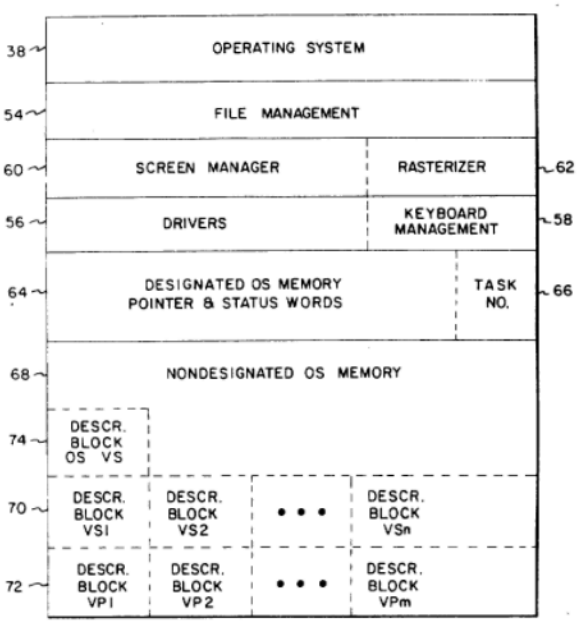
Claims of the '967 patent	Agarwal
	screen into viewport windows. Because each viewport is associated with an active task, the view ports are positioned side-by-side. [5:59-66] The file management system 54 manipulates data to and from the keyboard, disc storage and input /output unit. The file management system interfaces with the peripherals through drivers 56 which include the software required for interfacing with the specific peripherals used. [6:3-8] It can be seen that any number of virtual screens can be established by the screen manager in response to requests from application tasks and each virtual screen can be subdivided into any number of viewports by additional requests from the respective application task. Each virtual screen and each viewport is defined by a descriptor block which sets the size of the virtual screen or viewport, points to a page or document in the document files which is to be displayed in the virtual screen or viewport and sets the logical position of the screen or viewport relative to the page or document. [7:55-65] This controller handles the several functions described above. In particular, it implements the CREATE VIRTUAL SCREEN, CRE ATE VIEWPORT, ASSIGN, UPDATE, MERGE, AND DELETE functions with respect to particular descriptor data blocks 68. [10:3-8] The present technique also allows for the virtual screen of primary interest to make up a much larger portion of the physical screen. The use of the operating screen 52in the display gives the operating system sufficient opportunity to keep the user informed as to the status of virtual screens which are not displayed. Further, the ability of the operating system to establish viewports in each virtual screen greatly adds to the flexibility of the system, particularly with respect to displaying different types of data such as text and graphics. The information displayed in different viewports can also be selected from different pages and even different documents in the document files 48. The example of displaying text adjacent to graphics using the view port technique has previously been noted. Establishing viewport descriptor blocks for other items such as the menu and error messages of FIG. 2 makes the screen manager operations extremely flexible. It also minimizes the amount of updating of the screen. For

Claims of the ’967 patent	Agarwal
	example, in order to update the prompts viewport, which may re quire frequent updating, it is not necessary to as frequently update the entire screen. Similarly, when word processing, it may only be necessary to update the text viewport and not the other viewports at particular stages of an application task. The ability of the applications task to further subdivide pages into areas adds yet another dimension to the control of information to be displayed. It allows the application task to establish areas to be displayed in a relatively fixed relationship as far as the screen manager is concerned; whereas, the viewport technique requires the screen manager to handle each viewport more independently. Establishing areas simplifies certain tasks of the application software such as formatting, wrap around within columns and the like. [10:42-11:8]  *Fig. 1* [Figure 1]

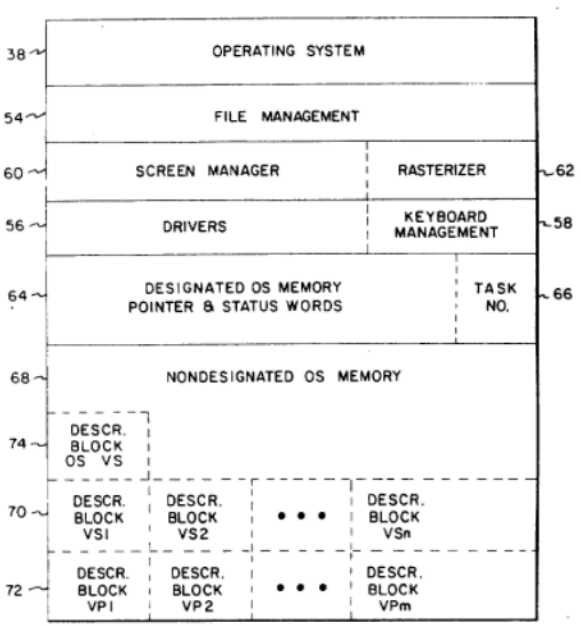
Claims of the '967 patent	Agarwal
	 *Fig. 8* [Figure 8]
Claim 2	
[2.1] The method of claim 1 wherein the data structure of the objects includes a header and one or more data segments	Agarwal discloses that the data structure of the objects includes a header and one or more data segments. *See* claims 1.3, 1.4, and 1.5. *See also, e.g.,:*

Claims of the ’967 patent	Agarwal
	 *Fig.4* [Figure 4] Preferably, the virtual screens are identified by descriptor data blocks stored by the screen manager. The descriptor data blocks designate portions of stored documents which are more directly handled by the application tasks and which are to be displayed. In response to requests by application tasks, the descriptor blocks are modified even when the virtual screens to which the blocks relate are held in background and thus not dis played. [2:12-20]

Claims of the '967 patent	Agarwal
	Distinct sections of document data stored in memory under control of the applications task are designated for each viewport. [2:38-40.] The data structures of the virtual screen and view 65 port descriptor blocks are shown in FIG. 5. A virtual screen description block includes a pointer to a first viewport descriptor block. The location and size of that first viewport corresponds to that of the entire virtual screen. If the primary viewport is subdivided into other viewports, each descriptor block of a subdivided view port points to the next in a series of viewport descriptor blocks linked by pointers. Each descriptor block includes the location and size of its respective subdivision. It is also given a viewport number by which it can be identified in requests from the application task. Each viewport descriptor block in the chain points to a page descriptor block in the document files. As also shown in FIG. 5 each page descriptor block defines the size of the page and indicates the position of a cursor within the page. [6:60-7:8]
[2.2] and wherein generating the second partition includes providing the first group of command functions with a first subgroup of command functions which are selectable to permit random movement between applications.	Agarwal discloses that generating the second partition includes providing the first group of command functions with a first subgroup of command functions which are selectable to permit random movement between applications. *See* claims 1.5 and 1.7. *See also, e.g.,:* A display from the foreground task is provided on a major portion of the physical display screen indicated as the task screen 50. Under control of the operating system to be described below, the display on the task screen may be divided into a number of display viewports each of which independently displays a different set of information. [4:62-68] It can be seen that the present system offers windowing at two levels. At a task level, in virtual screen windowing a task window covers virtually the entire physical screen. Within each virtual screen established by a particular application task, that task can subdivide the virtual screen into viewport windows. Because each viewport is associated with an active task, the view ports are positioned side-by-side. [5:59-66] It can be seen that any number of virtual screens can be established by the screen manager in response to requests from application tasks and each virtual screen can be subdivided into any

Claims of the ’967 patent	Agarwal
	number of viewports by additional requests from the respective application task. Each virtual screen and each viewport is defined by a descriptor block which sets the size of the virtual screen or viewport, points to a page or document in the document files which is to be displayed in the virtual screen or viewport and sets the logical position of the screen or viewport relative to the page or document. [7:55-65]
Claim 3	
[3.1] The method of claim 2 wherein the objects are stored at the respective reception systems in accordance with a predetermined plan	Agarwal discloses that the objects are stored at the respective reception systems in accordance with a predetermined plan. *See* claims 1.3, 1.4, and 1.5. *See also, e.g.,:*  *Fig.4*

Claims of the '967 patent	Agarwal
	[FIGURE 4]
[3.2] and wherein providing the first subgroup of commands includes providing a command for causing the user to be presented with at least one procedure for navigating to a new application.	Agarwal discloses that providing the first subgroup of commands includes providing a command for causing the user to be presented with at least one procedure for navigating to a new application. *See* claims 1.5 and 1.7. *See also, e.g.,:* A display from the foreground task is provided on a major portion of the physical display screen indicated as the task screen 50. Under control of the operating system to be described below, the display on the task screen may be divided into a number of display viewports each of which independently displays a different set of information. [4:62-68] It can be seen that the present system offers windowing at two levels. At a task level, in virtual screen windowing a task window covers virtually the entire physical screen. Within each virtual screen established by a particular application task, that task can subdivide the virtual screen into viewport windows. Because each viewport is associated with an active task, the view ports are positioned side-by-side. [5:59-66] It can be seen that any number of virtual screens can be established by the screen manager in response to requests from application tasks and each virtual screen can be subdivided into any number of viewports by additional requests from the respective application task. Each virtual screen and each viewport is defined by a descriptor block which sets the size of the virtual screen or viewport, points to a page or document in the document files which is to be displayed in the virtual screen or viewport and sets the logical position of the screen or viewport relative to the page or document. [7:55-65]
Claim 4	
[4.1] The method of claim 2 wherein the predetermined plan for storing objects at the respective reception systems includes providing the objects with a	Agarwal discloses that the predetermined plan for storing objects at the respective reception systems includes providing the objects with a storage control parameter in their respective headers. *See* claims 1.3, 1.4, 1.5, and 2.1. *See also, e.g.,:*

Claims of the '967 patent	Agarwal
storage control parameter in their respective headers,	 *Fig.4* [Figure 4] Additional prior art that discloses this limitation and is combinable with the present prior art is found in the relevant section of charts for other prior art for the '967 patent. It would have been obvious to one of ordinary skill in the art at the time of the invention to include such functionality in the system disclosed in Agarwal. *See, e.g.*, Preliminary Invalidity Contentions Section II.D.
[4.2] and wherein providing the first subgroup of command	Agarwal discloses that providing the first subgroup of command functions includes providing at least one command for causing the user to be presented with a plurality of different

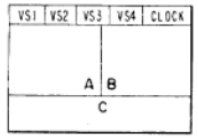
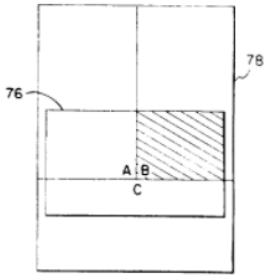
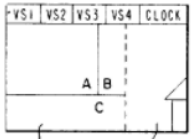
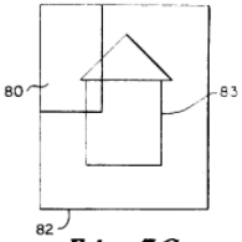
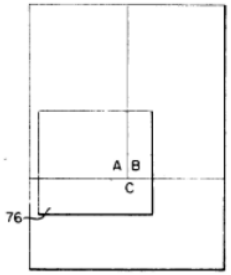
Claims of the '967 patent	Agarwal
functions includes providing at least one command for causing the user to be presented with a plurality of different procedure for navigating to a new application.	procedure for navigating to a new application. *See* claims 1.5 and 1.7. *See also, e.g.,:* A display from the foreground task is provided on a major portion of the physical display screen indicated as the task screen 50. Under control of the operating system to be described below, the display on the task screen may be divided into a number of display viewports each of which independently displays a different set of information. [4:62-68] It can be seen that the present system offers windowing at two levels. At a task level, in virtual screen windowing a task window covers virtually the entire physical screen. Within each virtual screen established by a particular application task, that task can subdivide the virtual screen into viewport windows. Because each viewport is associated with an active task, the view ports are positioned side-by-side. [5:59-66] It can be seen that any number of virtual screens can be established by the screen manager in response to requests from application tasks and each virtual screen can be subdivided into any number of viewports by additional requests from the respective application task. Each virtual screen and each viewport is defined by a descriptor block which sets the size of the virtual screen or viewport, points to a page or document in the document files which is to be displayed in the virtual screen or viewport and sets the logical position of the screen or viewport relative to the page or document. [7:55-65]
Claim 5	
[5.1] The method of claim 4 wherein the object storage control parameter is dependent on the currency of the object data,	Agarwal discloses that the object storage control parameter is dependent on the currency of the object data. *See* claims 1.1, 1.4, and 1.5. *See also, e.g.,:* At least one input/output unit 36 is also connected to the bus 24. The input /output unit 36 includes a communications port for communicating with a printer, other work stations or a main processing unit. Although the present invention is described with respect to a standalone word processing and office automation system, the invention is equally applicable to other systems such as distributed systems. [4:13-19]

Claims of the '967 patent	Agarwal
	Additional prior art that discloses this limitation and is combinable with the present prior art is found in the relevant section of charts for other prior art for the '967 patent. It would have been obvious to one of ordinary skill in the art at the time of the invention to include such functionality in the system disclosed in Agarwal. *See, e.g.,* Preliminary Invalidity Contentions Section II.D.
[5.2] and wherein providing the navigation procedures includes enabling the user to enter a character string at the reception system to randomly search the available applications for a desired application.	Agarwal discloses that providing the navigation procedures includes enabling the user to enter a character string at the reception system to randomly search the available applications for a desired application. *See* claims 1.5 and 1.7. *See also, e.g.,:* A display from the foreground task is provided on a major portion of the physical display screen indicated as the task screen 50. Under control of the operating system to be described below, the display on the task screen may be divided into a number of display viewports each of which independently displays a different set of information. [4:62-68] It can be seen that the present system offers windowing at two levels. At a task level, in virtual screen windowing a task window covers virtually the entire physical screen. Within each virtual screen established by a particular application task, that task can subdivide the virtual screen into viewport windows. Because each viewport is associated with an active task, the view ports are positioned side-by-side. [5:59-66] It can be seen that any number of virtual screens can be established by the screen manager in response to requests from application tasks and each virtual screen can be subdivided into any number of viewports by additional requests from the respective application task. Each virtual screen and each viewport is defined by a descriptor block which sets the size of the virtual screen or viewport, points to a page or document in the document files which is to be displayed in the virtual screen or viewport and sets the logical position of the screen or viewport relative to the page or document. [7:55-65]
Claim 6	
6. The method of claim 4 wherein	Agarwal discloses that providing the navigation procedures includes enabling the user to

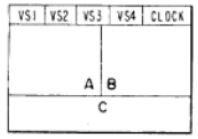
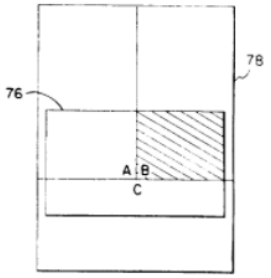
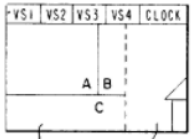
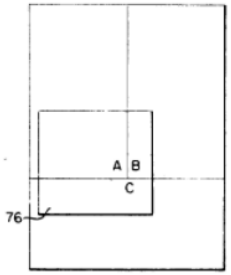
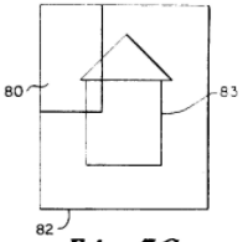
Claims of the '967 patent	Agarwal
providing the navigation procedures includes enabling the user to access an index of available applications from which a desired application may be selected.	access an index of available applications from which a desired application may be selected. *See* claims 1.5 and 1.7. *See also, e.g.,:* A display from the foreground task is provided on a major portion of the physical display screen indicated as the task screen 50. Under control of the operating system to be described below, the display on the task screen may be divided into a number of display viewports each of which independently displays a different set of information. [4:62-68] It can be seen that the present system offers windowing at two levels. At a task level, in virtual screen windowing a task window covers virtually the entire physical screen. Within each virtual screen established by a particular application task, that task can subdivide the virtual screen into viewport windows. Because each viewport is associated with an active task, the view ports are positioned side-by-side. [5:59-66] It can be seen that any number of virtual screens can be established by the screen manager in response to requests from application tasks and each virtual screen can be subdivided into any number of viewports by additional requests from the respective application task. Each virtual screen and each viewport is defined by a descriptor block which sets the size of the virtual screen or viewport, points to a page or document in the document files which is to be displayed in the virtual screen or viewport and sets the logical position of the screen or viewport relative to the page or document. [7:55-65]
Claim 7	
7. The method of claim 4 wherein providing the navigation procedures includes enabling the user to access a directory of application subject matter from which a desired application may be selected.	Agarwal discloses that providing the navigation procedures includes enabling the user to access a directory of application subject matter from which a desired application may be selected. *See* claims 1.5 and 1.7. *See also, e.g.,:* A display from the foreground task is provided on a major portion of the physical display screen indicated as the task screen 50. Under control of the operating system to be described below, the display on the task screen may be divided into a number of display viewports each of which independently displays a different set of information. [4:62-68]

Claims of the '967 patent	Agarwal
	It can be seen that the present system offers windowing at two levels. At a task level, in virtual screen windowing a task window covers virtually the entire physical screen. Within each virtual screen established by a particular application task, that task can subdivide the virtual screen into viewport windows. Because each viewport is associated with an active task, the view ports are positioned side-by-side. [5:59-66] It can be seen that any number of virtual screens can be established by the screen manager in response to requests from application tasks and each virtual screen can be subdivided into any number of viewports by additional requests from the respective application task. Each virtual screen and each viewport is defined by a descriptor block which sets the size of the virtual screen or viewport, points to a page or document in the document files which is to be displayed in the virtual screen or viewport and sets the logical position of the screen or viewport relative to the page or document. [7:55-65]
Claim 8	
8. The method of claim 4 wherein providing the navigation procedures includes enabling the user to access a physical analogy of the available applications from which a desired application may be selected.	Agarwal discloses that providing the navigation procedures includes enabling the user to access a physical analogy of the available applications from which a desired application may be selected. *See* claim 1.1.
Claim 9	
9. The method of claims 5, 6, 7 or 8 wherein providing the navigation procedures to a new application includes presenting a window at the display in which the user is presented with multiple, interactive command functions to	Agarwal discloses that providing the navigation procedures to a new application includes presenting a window at the display in which the user is presented with multiple, interactive command functions to effect navigation. *See* claims 1.5 and 1.7. *See also, e.g.,:* A display from the foreground task is provided on a major portion of the physical display screen indicated as the task screen 50. Under control of the operating system to be described below, the display on the task screen may be divided into a number of display viewports each

Claims of the '967 patent	Agarwal
effect navigation.	of which independently displays a different set of information. [4:62-68] It can be seen that the present system offers windowing at two levels. At a task level, in virtual screen windowing a task window covers virtually the entire physical screen. Within each virtual screen established by a particular application task, that task can subdivide the virtual screen into viewport windows. Because each viewport is associated with an active task, the view ports are positioned side-by-side. [5:59-66] It can be seen that any number of virtual screens can be established by the screen manager in response to requests from application tasks and each virtual screen can be subdivided into any number of viewports by additional requests from the respective application task. Each virtual screen and each viewport is defined by a descriptor block which sets the size of the virtual screen or viewport, points to a page or document in the document files which is to be displayed in the virtual screen or viewport and sets the logical position of the screen or viewport relative to the page or document. [7:55-65]
Claim 12	
[12.1] The method of claim 1 further including generating at least a third screen partition concurrently with the first and second screen partitions for presenting a second application and	Agarwal discloses including generating at least a third screen partition concurrently with the first and second screen partitions for presenting a second application. *See* claims 1.5 and 1.7. *See also, e.g.,:*

Claims of the ’967 patent	Agarwal
	 *Fig.6A*  *Fig.6B*  *Fig.7A*  *Fig.7C*  *Fig.7B* [Figures 6-7]
[12.2] wherein the data structure of the objects includes a header and one or more data segments.	Agarwal discloses that the data structure of the objects includes a header and one or more data segments. *See* claim 2.1.
Claim 13	

Claims of the '967 patent	Agarwal
[13.1] The method of claim 12 wherein the objects are stored at the respective reception systems in accordance with a predetermined plan	Agarwal discloses that the objects are stored at the respective reception systems in accordance with a predetermined plan. *See* claim 3.1.
[13.2] and wherein the predetermined plan for storing objects at the respective reception systems includes providing the objects with a storage control parameter in their respective headers,	Agarwal discloses that he predetermined plan for storing objects at the respective reception systems includes providing the objects with a storage control parameter in their respective headers. *See* claim 4.1.
[13.3] and wherein presenting a third screen partition includes presenting the second application as advertising.	Agarwal discloses that presenting a third screen partition includes presenting the second application as advertising. *See* claims 1.1, 1.4, 1.5, and 1.6. Additional prior art that discloses this limitation and is combinable with the present prior art is found in the relevant section of charts for other prior art for the '967 patent. It would have been obvious to one of ordinary skill in the art at the time of the invention to include such functionality in the system disclosed in Agarwal. *See, e.g.*, Preliminary Invalidity Contentions Section II.D.
Claim 14	
[14.1] The method of claim 1 further including generating one or more window partitions that overlays at least a portion of the application partition, the one or more windows for presenting data associated with the application displayed	Agarwal discloses generating one or more window partitions that overlays at least a portion of the application partition, the one or more windows for presenting data associated with the application displayed. *See* claims 1.5 and 1.7. *See also, e.g.,:* In conventional windowing techniques, several displays corresponding to the virtual screens of the present application are overlapped but spatially offset from each other on the physical screen. [10:29-32]

Claims of the ’967 patent	Agarwal
	 *Fig.6A*  *Fig.6B*  *Fig.7A*  *Fig.7B*  *Fig.7C* [Figures 6-7]
[14.2] and wherein the data structure of the objects includes a header and one or more data segments,	Agarwal discloses that the data structure of the objects includes a header and one or more data segments. *See* claim 2.1.

Claims of the '967 patent	Agarwal
[14.3] and wherein the objects are stored at the respective reception systems in accordance with a predetermined plan	Agarwal discloses that the objects are stored at the respective reception systems in accordance with a predetermined plan. *See* claim 3.1.
[14.4] which includes providing the objects with a storage control parameter at their respective headers.	Agarwal discloses providing the objects with a storage control parameter at their respective headers. *See* claim 4.1.
Claim 15	
15. The method of claim 14 wherein generating window partitions includes providing the window partitions with fields for conducting interactive procedures associated with an underlying application.	Agarwal discloses that generating window partitions includes providing the window partitions with fields for conducting interactive procedures associated with an underlying application. *See* claims 1.5 and 1.6. *See also, e.g.,:* The system of FIG. 1 is a multi-tasking system. That is, the CPU is able to process several application tasks together in a multiplexed fashion. However, as will be described in greater detail below, the system user interacts with only one of those tasks at a time through the display 32 and keyboard 28. For that one task, which is the foreground task, the user may enter text data and text/document manipulation commands by means of keystrokes through the keyboard 28. The work station responds by executing in the CPU 22 the appropriate routines selected by the operating system 38 and, through the operating system, by the applications task 46. In executing those routines, the CPU may modify the contents of documents in the document files 48 and display results of the user input through the display 32. [4:46-60]
Claim 16	
16. The method of claim 15 wherein generating window partitions includes providing the window partitions with interactive fields for conducting transactional procedures.	Agarwal discloses that generating window partitions includes providing the window partitions with interactive fields for conducting transactional procedures. *See* claim 15.
Claim 17	

Claims of the '967 patent	Agarwal
[17.1] The method of claim 1 wherein generating the first and second screen partitions includes generating the respective partitions at fixed, predetermined regions of the display screen, the second partition being arranged as a command bar	Agarwal discloses that generating the first and second screen partitions includes generating the respective partitions at fixed, predetermined regions of the display screen, the second partition being arranged as a command bar. *See* claims 1.5 and 1.7. *See also, e.g.,:* This controller handles the several functions described above. In particular, it implements the CREATE VIRTUAL SCREEN, CRE ATE VIEWPORT, ASSIGN, UPDATE, MERGE, AND DELETE functions with respect to particular descriptor data blocks 68. [10:3-8] The example of displaying text adjacent to graphics using the view port technique has previously been noted. Establishing viewport descriptor blocks for other items such as the menu and error messages of FIG. 2 makes the screen manager operations extremely flexible. [10:42-11:8]
[17.2] and wherein the data structure of the objects includes a header and one or more data segments, and	Agarwal discloses that the data structure of the objects includes a header and one or more data segments. *See* claim 2.1.
[17.3] wherein the objects are stored at the respective reception systems in accordance with a predetermined plan,	Agarwal discloses that the objects are stored at the respective reception systems in accordance with a predetermined plan. *See* claim 3.1.
[17.4] which includes providing the objects with a storage control parameter at their respective headers.	Agarwal discloses providing the objects with a storage control parameter at their respective headers. *See* claim 4.1.